

Rancang Bangun Sistem Kendali *Compliant Motion* Pada Robot Industri

Indrawanto^{1,*}, Muhammad Ilham Nafan² dan Miftakhudin³

^{1,2,3}Institut Teknologi Bandung, Jl. Ganesha 10 Bandung, Indonesia

*indrawanto@ftmd.itb.ac.id

Abstrak

Dalam makalah ini akan disajikan hasil penelitian rancang bangun sistem kendali *compliant motion* untuk tracking kontur permukaan menggunakan robot industri Yaskawa Motoman MH80. Untuk mengukur gaya kontak yang terjadi antara *end-effector* robot dengan permukaan kontur digunakan sensor gaya 6 DOF. Gaya kontak yang terukur digunakan sebagai umpan balik sistem kendali untuk mengendalikan gerak robot pada arah tertentu sambil mempertahankan gaya kontak yang terjadi antara *end effector* robot dengan permukaan pada nilai tertentu. Dari hasil percobaan didapatkan bahwa *end effector* robot dapat melakukan *tracking* permukaan dengan mempertahankan gaya tertentu terhadap profil sebuah permukaan dengan baik.

Kata kunci : robot industri, *compliant motion*, sensor gaya, *tracking*, sistem kendali

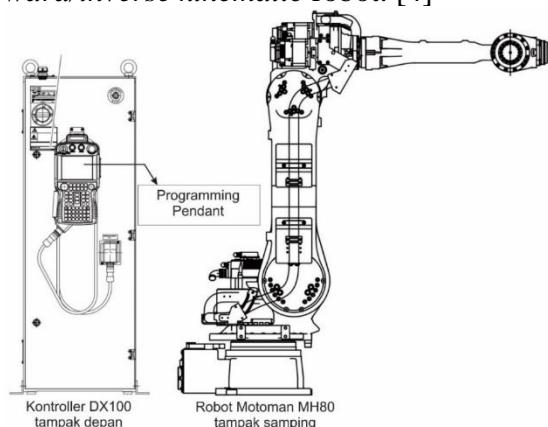
Pendahuluan

Compliant Motion adalah gerak dimana *end-effector* robot menyentuh lingkungan atau permukaan saat robot bergerak. Untuk mengendalikan *compliant motion* ini umumnya dapat dilakukan dengan dua pendekatan yakni *impedance control* dan *hybrid position-force control* [1-3]. Pada *impedance control* pengendalian dilakukan dengan mengatur kekakuan robot pada arah dimana terjadi kontak sedangkan pada *hybrid position-force control* pengendalian dilakukan dengan mengukur gaya kontak yang terjadi lalu mengatur gerak dengan memisahkan arah yang dikendalikan berdasarkan perintah gaya dan arah dengan kendali posisi. Penggunaan *compliant motion* pada industry misal untuk pemolesan, *deburring* dsb.

Proses *tracking* kontur pada makalah ini dilakukan dengan menggunakan teknik pengendalian berbasis proporsional. Sistem yang digunakan hanya berbasis

proporsional dikarenakan pengendalian integral sudah terdapat pada pengendali robot.

Proses *tracking* menggunakan Robot Yaskawa. Robot industri dengan tipe MH80 tersebut diatur oleh DX100 yang di dalamnya mengendalikan enam buah motor AC servo, pengolah data *encoder*, dan *forward/inverse kinematic* robot. [4]



Gambar 1. Bagian-bagian robot

Robot Yaskawa Motoman memberikan beberapa metode pengendalian yaitu [5]:

- Incremental move

- Linear move

Incremental move yaitu perintah gerak *end effector* yang tidak menggunakan titik referensi tetap melainkan titik referensi dalam pergerakan adalah posisi end effector pada waktu akan dilakukan perintah gerak itu sendiri. Sedangkan *linear moving* yaitu *end effector* bergerak menuju koordinat posisi dan orientasi yang diinginkan relatif terhadap titik acuan tertentu yang bersifat *fixed*. Selain itu digunakan juga sensor.

Pada pengendalian ini umpan balik berupa sensor gaya dan torsi digunakan Sensor ROBOTIQ FT 150 [6] yang dapat mengukur gaya dan torsi untuk sumbu *x*, *y* dan *z*.

Dari robot dan sensor tersebut dilakukan perhitungan pengendalian dengan menggunakan komputer. Komputer ini selain berfungsi sebagai perhitungan pengendalian juga berfungsi sebagai GUI atau yang lebih dikenal dengan *Graphical User Interface* menggunakan Microsoft Visual Studio C#. GUI ini merupakan perantara antara operator dengan komputer untuk dapat berinteraksi.

Perancangan dan Pembuatan Kendali

Dalam tahapan ini untuk membuat sistem kendali terlebih dahulu dilakukan persiapan sistem komunikasi robot dan melakukan komunikasi pada sensor gaya dan torsi agar dapat beroprasi software yang terdapat pada komputer.

Persiapan sistem sensor gaya dan torsi. Sistem komunikasi yang digunakan antara komputer dengan sensor adalah ModBus RTU dengan mekanisme timbal balik via kabel RS485. Dengan arti apabila diberikan pertintaan permintaan barulah sensor memberi nilai yang dibaca pada saat itu. Tabel 1 berikut ini adalah kode komunikasi permintaan yang dikirim ke sensor:

09 03 00 B4 00 09 C4 A2

Tabel 15. Tabel penjabaran perintah permintaan modbus RTU

Bits	Deskripsi
09	SlaveID
03	Function Code 02 (Membaca <i>Holding Registers</i>)
00 B4	Alamat register awal yang di minta
00 09	Jumlah register yang di minta (9)
C4 A2	<i>Cyclic Redundancy Check (CRC)</i>

Respon yang diberikan dari sensor yang dijelaskan pada Tabel 2:

09 03 12 04 D7 FA 47 F7 05 05 BA 01 3C FB 36
D9 F6

Tabel 16. Tabel Penjabaran Respon Modbus

Bits	Deskripsi
09	SlaveID
03	Function Code 02 (Membaca <i>Holding Registers</i>)
12	Jumlah byte data yang akan digunakan (6 registers x 2 bytes/register = 12 bytes)
04 D7	Nilai register 00B4 (04D7 = 1239, $F_x = 12,39 \text{ N}$)
FA 47	Nilai register 00B5 (FA47 = -1465, $F_y = -14,65 \text{ N}$)
F7 05	Nilai register 00B6 (F705 = -2299, $F_z = -22,99 \text{ N}$)
05 BA	Nilai register 00B7 (05BA = 1466, $R_x = 1,466 \text{ Nm}$)
01 3C	Nilai register 00B8 (013C = 316, $R_y = 0,316 \text{ Nm}$)
FB 36	Nilai register 00B9 (FB36 = -1226, $R_z = -1,226 \text{ Nm}$)
D9 F6	<i>Cyclic Redundancy Check (CRC)</i>

Penentuan matriks kalibrasi. Tujuan dari kalibrasi adalah pengukuran yaitu membandingan dengan suatu besaran standar yang menjadi acuan kalibrasi. Diperlukan matriks kalibrasi yang merupakan sebuah matriks yang berfungsi untuk mengkompensasi keluaran sensor sesuai dengan hasil kalibrasi. Perhitungan yang dilakukan dapat dilihat pada Pers. 1 berikut:

$$[F'] = [C][F] \quad (1)$$

Matriks $[C]$ menghubungkan matriks beban $[F_x]$ dengan nilai keluaran dari sensor $[F'_x]$. Apabila nilai yang diberikan adalah terhadap sumbu x, y dan z, maka matriks yang digunakan berdimensi 3×3 untuk F dan matriks yang sama untuk R (momen) untuk persamaan F dapat dilihat pada Pers. 2 di bawah ini:

$$\begin{bmatrix} F'_x \\ F'_y \\ F'_z \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (2)$$

Matriks kalibrasi $[K]$ didapat dengan cara melakukan operasi pada Pers. 1 sebagai berikut

$$[F] = [C]^{-1}[F'] = [K][F'] \quad (3)$$

Peengkalbrasian matriks F dilakukan dengan memberi beban secara meningkat dan menurun dengan jarak setiap 2 kg dengan massa hingga 14 kg untuk tiap-tiap sumbu. Sedangkan matriks R dilakukan dengan metode yang sama hingga 1 Nm. Dari pengujian didapat dapat matriks kalibrasi F sebagai berikut, Pers. 4

$$K1 = C1^{-1} = \begin{bmatrix} 1,0255 & -0,1147 & -0,0730 \\ -0,1725 & 1,0084 & -0,0705 \\ -0,0201 & -0,0223 & 1,0365 \end{bmatrix} \quad (4)$$

Sedangkan, matriks kalibrasi R ditunjukkan pada Pers. 5:

$$K2 = C2^{-1} = \begin{bmatrix} 1,4221 & -0,5350 & -0,0119 \\ -0,3667 & 1,4626 & -0,0188 \\ -0,3330 & -0,5809 & 0,9872 \end{bmatrix} \quad (5)$$

Komunikasi Perintah Pada Robot.
Komunikasi robot dengan komputer dilakukan dengan metode TCP/IP dimana pengendali robot dalam hal ini berperan sebagai server yang apabila diberikan

perintah tertentu akan dilakukan dengan output yang berkenaan pada kendali robot. Sehingga dibuat *software* yang berfungsi untuk mengatur robot via komputer. Perlu dikeahui bahwa untuk perintah yang diberikan robot yaitu berupa perintah gerak dengan *output* berupa posisi sedangkan perhitungan kinematik robot sepenuhnya dilakukan oleh pengendali.

Dari hasil referensi yaitu perintah gerak secara *incremental* dan linier *move* didapat dengan mengirim data dengan format berikut, Tabel 2):

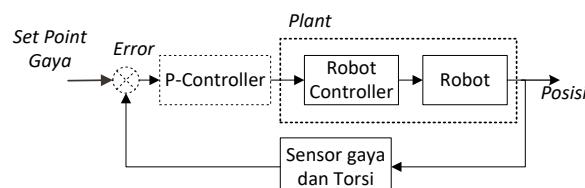
Tabel 2. Perintah Tabel Sensor

```
"cmd=42;sCtrlGrp=0;sVTyp
e=0;lSpeed=" + tbSpeed.Text +
";sFrame=1;sConfig=0;sTo
o1No=0;lPosA=" + tbMovX.Text + ";lPosB=" +
+ tbMovY.Text + ";lPosC=" + tbMovZ.Text +
+ ";lPosD=" + tbMovRX.Text + ";lPosE=" +
+ tbMovRY.Text + ";lPosF=" + tbMovRZ.Text +
+
";lPosG=0;lPosH=0;lPosI=
0;lPosJ=0;lPosK=0;lPosL=
0;"
```

Cmd =42 merupakan perintah *incremental moving* dan cmd = 40 untuk liner *moving*. Parameter yang dapat dikendalikan adalah yang ditulis teks *italic* (miring) yaitu kecepatan, posisi Cartesian dan orientasi yang masing-masing terhadap sumbu x,y,z .

Selain itu terdapat juga perintah cmd=15 yang berfungsi untuk meminta status posisi *end effector* sehingga posisi dapat diketahui secara *realtime*.

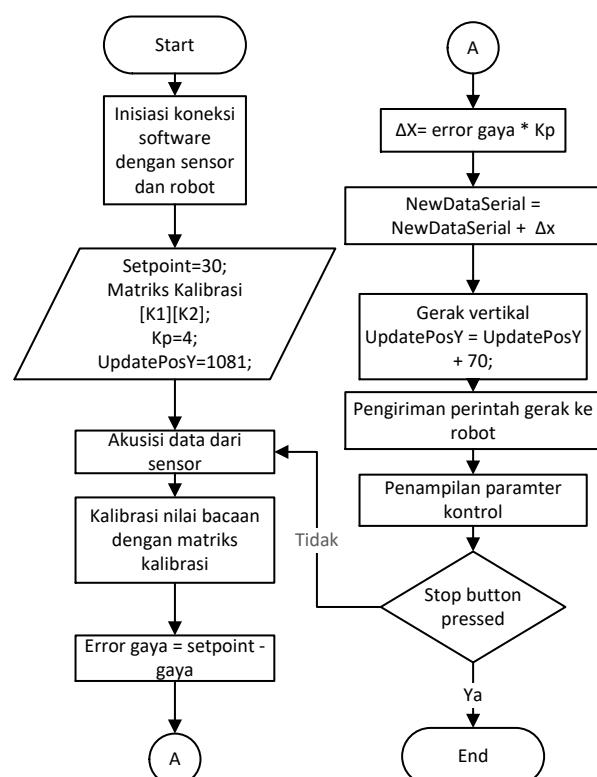
Perancangan Algoritma Pengendali
Perancangan algoritma yang akan digunakan untuk *tracking* kontur dikembangkan berdasarkan diagram blok dibawah ini, Gambar 2:



Gambar 2. Diagram blok pengendali

Proses perhitungan untuk pengendali proporsional tersebut dilakukan pada *software*. Pada penelitian ini dibuat beberapa algoritma yang bertujuan untuk mencari kemampuan terbaik tracking sebuah kontur.

- Algoritma Proporsional merupakan inti dari pengendali *tracking* kontur berbasis proporsional yang ditunjukkan pada Gambar 3. Nilai K_p yang dipilih secara percobaan adalah 4 dengan setpoint gaya 30 N.

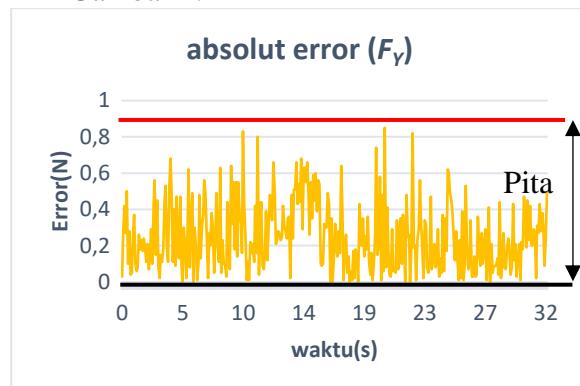


Gambar 3. Diagram alir algoritma kendali proporsional

- Algoritma Pita (*Band*) berfungsi untuk mengkompensasi nilai *error* yang berubah-ubah dibawah rentang nilai yang ditetapkan sehingga *end*

effector tidak langsung bergerak dikarenakan gaya yang relatif kecil. Selain itu juga untuk meminimalisir pengaruh terjadinya pristiwa *noise*. Berikut ini penjelasan mengenai pita pada algoritma ini berupa grafik *error* absolut sensor pada sumbu F_y

Gambar 4.

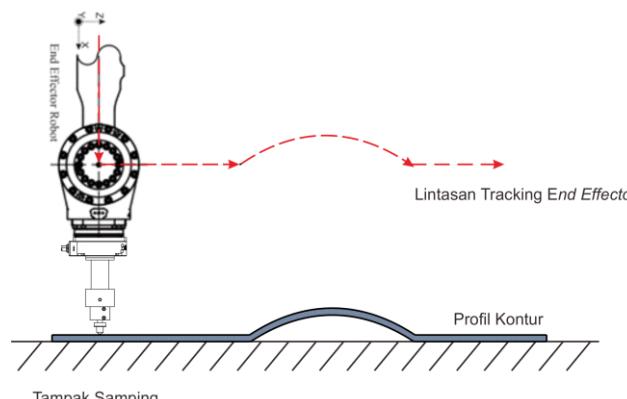


Gambar 4. Skema pita

Apabila nilai *error* absolut tersebut belum keluar dari pita maka dianggap tidak terjadi perubahan pada input.

Tracking End Effector. Proses ini bertujuan untuk menguji sistem kendali yang telah dikembangkan. Pengujian dapat dilihat pada gambar 4. Pada pengujian ini hanya dilakukan proses *tracking* dengan orientasi *end effector* tetap. Dalam proses *tracking* nilai *gain* K_p yang digunakan yaitu bernilai 4 dan digunakan *setpoint* 30 N.

- Algoritma Penunggu adalah



algoritma untuk membuat perintah gerak yang di berikan kepada robot

agar tidak terus menerus dilakukan. Melainkan, pemberian perintah gerak selanjutnya menunggu hingga robot tersebut selesai menjalankan perintah gerak dari perintah sebelumnya.

Dari ketiga algoritma tersebut diaplikasikan pada *software* pengendali yang akan digunakan pada proses *tracking* kontur.

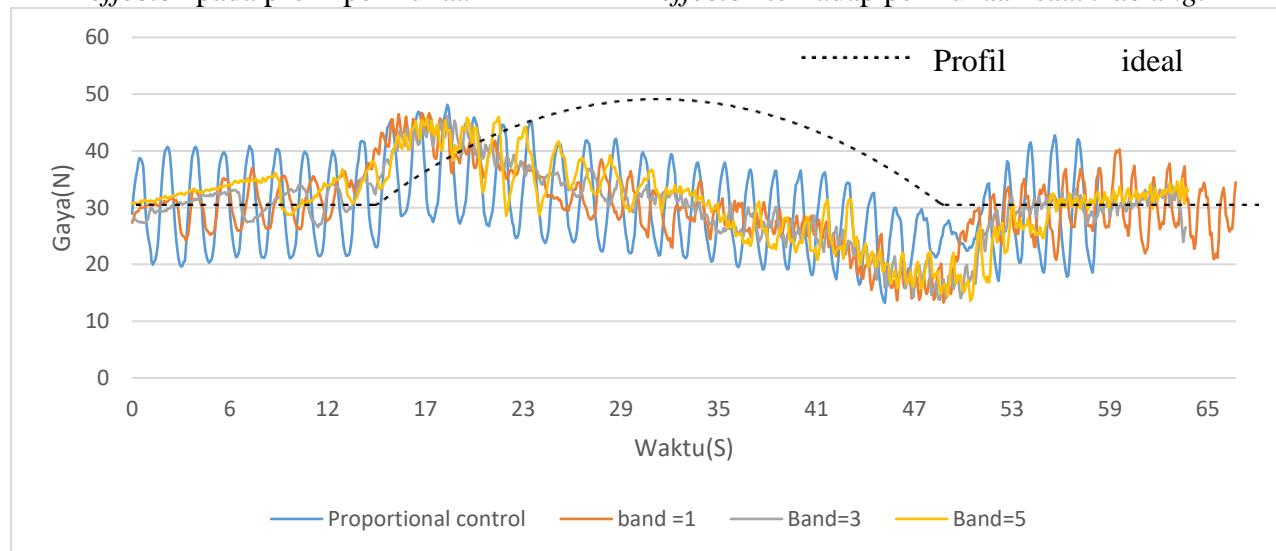
Pengujian dan Analisis

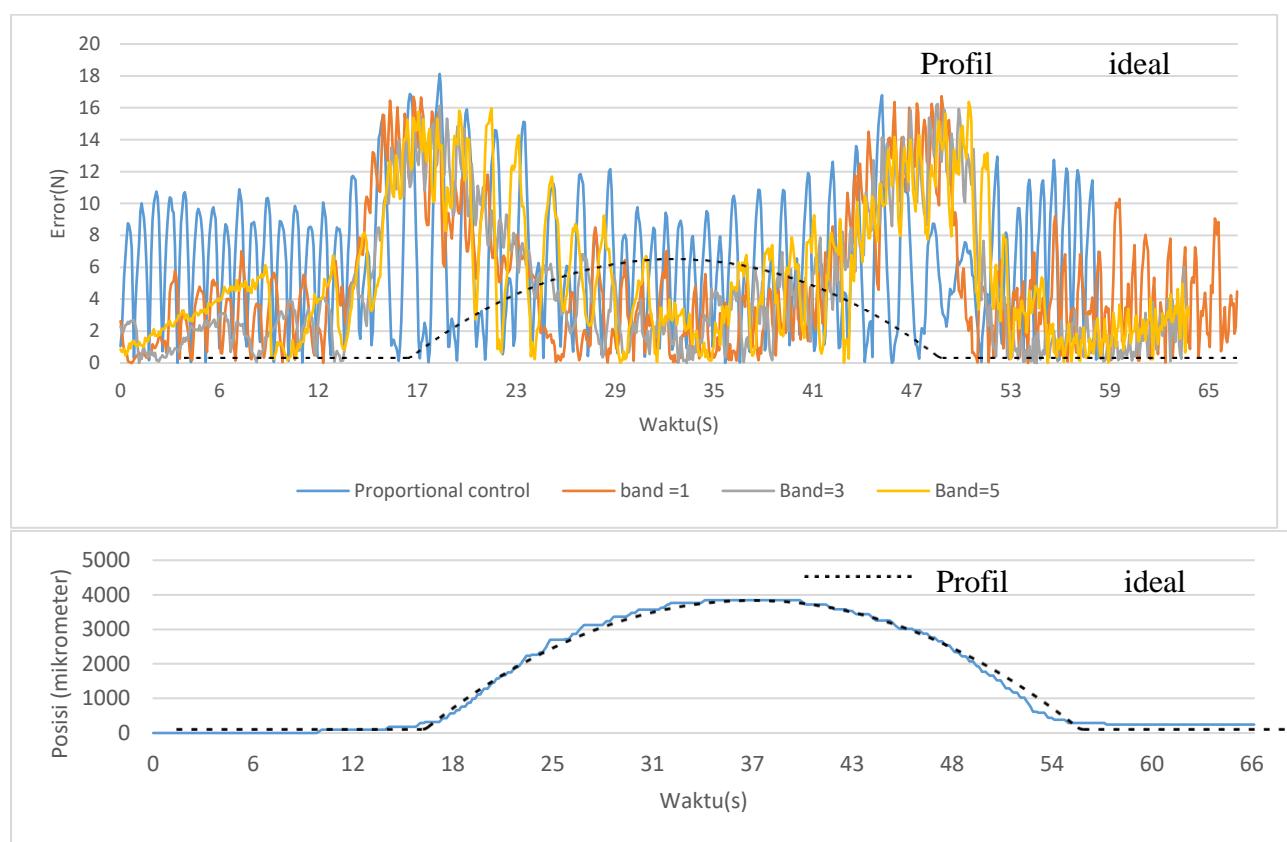
Pengujian dilakukan dengan nilai band yang berbeda-beda yaitu band satu, tiga dan lima untuk melihat pengaruh respon pengontrolan terhadap parameter band ini.

Gambar 4. Skema pengujian *tracking end effector* pada profil permukaan



Hasil pengujian *tracking end effector* pada profil permukaan ditunjukkan pada Gambar 5. Pada pengujian ini robot bergerak ke satu arah sambil mempertahankan gaya kontak antara *end effector* terhadap permukaan saat *tracking*.





Gambar 5. Grafik hasil proses tracking kontur berupa keluaran gaya (atas); error absolut (tengah) dan kurva permukaan dan posisi *end effector* vs waktu (bawah).

Kesimpulan

Berdasarkan hasil pengujian *tracking* kontur ini didapat bahwa penggunaan algoritma kendali proporsional terjadi osilasi yang relatif tinggi dibandingkan pengendalian menggunakan algoritma penunggu dan pita. Untuk pengendalian menggunakan algoritma dengan nilai band yang berbeda-beda, didapat bahwa nilai band yang menghasilkan variasi paling kecil yaitu dengan besar band 3. Hasil pengamatan nilai error tertinggi secara umum terjadi pada saat ujung batang *end effector* mencapai wilayah naik ataupun turun bukit dari profil yang telah dibuat dengan nilai error terbesar bernilai 18N. Untuk menurunkan nilai error tersebut direncanakan untuk mendeteksi gradient

permukaan kontur sebagai kompensasi perintah kendali.

Referensi

- [1] Mason, M. "Compliance and Force Control for Computed Controlled Manipulators," M.S. Thesis, MIT AI Laboratory, May 1978
- [2] Craig, J. dan Raibert, M. "A Systematic Method for Hybrid Position/Force Control of a Manipulator," Proceedings of the 1979 IEEE Computer Software Applications Conference, Chicago, November 1979
- [3] Raibert, M. dan Craig, J. "Hybrid Position/Force Control of Manipulators," ASME Journal of Dynamic Systems, Measurement, and Control, June 1981

[4] Adrianto, I. P. (2015). Aplikasi Program MOTOPLUSIDE® Pada Pengendalian Waktu Riil Robot Industri Yaskawa Motoman MH80. Bandung: ITB.

[5] YASKAWA MOTOMAN Manuals:
<http://www motoman com/motomedia/manuals/docs/158813-1CD-R1.pdf> (Diakses 20 April 2016).

[6] Robotiq Force Torque Sensor FT150 Specifications, <http://robotiq.com/wp-content/uploads/2014/09/Robotiq-Force-Torque-Sensor-FT-150-Specifications.pdf> (Diakses 20 Mei 2016)